



mQMA: multi-constrained QoS Multicast Aggregation

Naouel Ben Ali, Joanna Moulhierac, Abdelfattah Belghith, Miklos Molnar

► To cite this version:

Naouel Ben Ali, Joanna Moulhierac, Abdelfattah Belghith, Miklos Molnar. mQMA: multi-constrained QoS Multicast Aggregation. Globecom, IEEE, Nov 2007, Washington, United States. inria-00428580

HAL Id: inria-00428580

<https://inria.hal.science/inria-00428580>

Submitted on 29 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

mQMA: multi-constrained QoS Multicast Aggregation

Naouel Ben Ali[†], Joanna Moulrierac^{*}, Abdelfattah Belghith[†] and Miklós Molnár^{*}

^{*} IRISA, University of Rennes I, France

[†] CRISTAL, ENSI, Tunis, Tunisia

Email: naouel.benali@ensi.rnu.tn, joanna.moulrierac@irisa.fr,
abdefattah.belghith@ensi.rnu.tn and miklos.molnar@irisa.fr

Abstract—Traditional IP Multicast has been proposed in order to manage group communications over the Internet in a bandwidth efficient manner. Although this proposition has been well studied, there are still some problems for its deployment. In this paper, we propose a new algorithm mQMA that deals with two important problems of traditional IP multicast, *i.e.*, multicast forwarding state scalability and multi-constrained QoS routing. The algorithm mQMA builds few trees and maintains few forwarding states for the groups thanks to the technique of multicast tree aggregation, which allows several groups to share the same delivery tree. Moreover, the algorithm mQMA builds trees satisfying multiple QoS constraints. We show, through extensive simulations, that mQMA leverages the same QoS performances as Mamcra which is the main multi-constrained multicast routing algorithm. Moreover, mQMA reduces dramatically the number of trees to be maintained.

I. INTRODUCTION

Recently, the Internet has shown a tremendous growth. Emergent multimedia applications like audio/video conferencing, video on demand, IP-telephony usually have other requirements than traditional Internet services such as e-mail and file transfer services. IP multicast techniques have been proposed to support group communications over the Internet with the aim of reducing the network resource consumption. Although it has been studied for a long time, IP multicast is not well deployed over the Internet. The multicast forwarding state scalability problem and the lack of QoS support can be considered among the reasons for IP multicast not to be widely deployed.

Multicast tree aggregation [1], [2] is a recent approach that deals with the problem of multicast forwarding state scalability. In this approach, multiple groups share the same delivery tree within a domain whereas in traditional IP multicast, a separate tree is built and maintained for each group. With the multicast tree aggregation, fewer trees are maintained which reduces both the number of forwarding states in routers and the overhead induced by multicast control messages. Some information has to be added in the border routers of the domain in order to multiplex the data for the groups onto an aggregated tree. Several algorithms have been proposed to perform tree aggregation: AM [1] (Aggregated Multicast), STA [2] (Scalable Tree Aggregation) are the first proposed, BEAM [3] deals with distributed multicast tree aggregation, AMBTS [4] and TALD [5] deal specifically with tree aggregation in large

domains.

However, these algorithms aggregate multicast groups without considering any QoS requirements. Aggregation is based on trees computed by IP multicast routing protocols which use the shortest path tree algorithm optimized on one single metric, typically the hop count. Two solutions have been proposed in the literature to deal with QoS multicast aggregation: AQoS [6] and Q-STA [7]. The goal of these two algorithms is to aggregate groups to trees while respecting bandwidth requirements of groups. However, multicast applications today need to satisfy more than one or two QoS criteria that's why multiple QoS multicast aggregation is needed. Tree shared by multicast members must fulfill flow requirements such as delay, bandwidth, delay variation (also known as jitter), packet loss that can be tolerated and/or number of hops. To achieve QoS multicast aggregation under several QoS constraints, the forwarding structure must be computed by multi-constrained multicast routing algorithms.

In this paper, we propose a new algorithm called mQMA (multi-constrained QoS Multicast Aggregation) which performs multicast aggregation taking into account multiple QoS requirements. To the best of our knowledge, it is the first proposed protocol that deals with these two important problems of multicast forwarding state scalability and of multi-constrained multicast routing. Our protocol mQMA reduces the number of trees to be maintained and yet it builds trees satisfying several additive constraints.

The rest of the paper is organized as follows. Section II sets up the needed requirements by presenting the previous work that dealt with multicast tree aggregation and multi-constrained multicast routing. Section III describes our proposed algorithm mQMA. Section IV shows how mQMA algorithm improves the multi-constrained routing structure. Section V presents and analyzes the simulation results. Finally, we present some concluding remarks in section VI.

II. RELATED WORK

This section describes first the work related to multicast aggregation and then that related to multi-constrained multicast QoS routing.

A. Multicast Tree Aggregation

Multicast tree aggregation is proposed as a solution to the multicast forwarding state scalability problem. It is a multicast scheme which forces multicast groups to share the same delivery tree within a domain. In aggregated multicast, arriving groups must be matched to aggregated trees. The group-tree matching can be performed in different ways : the match is perfect when the aggregated tree covers exactly all the members of the arriving group and has a cost (in terms of bandwidth consumption) not larger than that of the native tree (computed by common multicast routing). When bandwidth waste is allowed, the leaky aggregation may associate a group to a higher cost tree. In this case, a bandwidth loss threshold must be specified, and if the bandwidth consumption of a tree exceeds the specified threshold, then it is discarded and not considered as an aggregated tree for the group. The authors in [9] studied the number of aggregated trees to be configured depending on the network topology and on the multicast routing algorithm. When aggregation is centralized, a tree manager stores the information concerning the group memberships and the trees maintained in the domain. The multicast Tree Set (*MTS*) contains all the multicast trees configured into the domain. So, whenever a border router receives a IGMP join or leave message for a group, it contacts the tree manager which is in charge of finding an adequate tree for the given group. In this phase, if the tree manager finds a tree respecting the bandwidth threshold, then the aggregation is performed. If not, the tree manager configures a brand new tree for the group. This tree becomes in turn a candidate for further aggregations. In all the cases, the tree manager informs the border router of the matching group-tree. Then, the border router will be able to route the packets for multicast groups using aggregated trees.

B. QoS Multicast Routing

The problem of QoS routing, even in the unicast case, is known to be NP-complete and has been extensively studied by the research community [10]. Before presenting the mechanisms used to construct multi-constrained QoS multicast delivery structure, we first specify some hypothesis used to solve this problem and the notation used throughout this paper. Then we present, rather briefly, the main existing approaches to achieve multicast QoS routing.

a) Hypothesis: QoS routing approaches assume that the network-state information is temporarily static and has been distributed (known) throughout the network. This network-state information is accurately maintained at each node and we assume that it can be collected by any appropriate traffic engineering mechanism. The QoS metrics are categorized into additive (e.g., delay, jitter,...) and min (resp. max) metrics (e.g. bandwidth). The constraints on min (resp. max) QoS measures can easily be treated by pruning all links (and disconnected nodes) which do not satisfy the requested min (resp. max) QoS constraints. In contrast, constraints on additive QoS measures cause more difficulties. We consider, hereafter and without loss of generality, that all QoS measures are additive.

A network topology is modeled as an undirected and valued graph $G = (V, E)$, where V is the set nodes and E the set of links. Each link is characterized by an m -dimensional link weight vector of m non-negative QoS weights (w_i^j , for $i = 1, 2, \dots, m$ and j is a link in E). The m QoS constraints (limits) L_i for $i = 1, 2, \dots, m$ are represented by the constraint vector. A multicast group g is composed of a source s and a set of members $D = \{d_1, d_2, \dots, d_k\}$ where k is the number of multicast members.

b) Multi-constrained multicast routing proposals: For the multicast case, a number of QoS routing algorithms based on single, dual and multiple metrics have been proposed. Single and dual metric QoS multicast routing algorithms have been proposed for cost, delay and the combinations: cost-delay and delay-jitter [8]. Few works have dealt with multiple QoS metrics for multicast routing. One way to tackle the multicast QoS routing problem is to compute a set of unicast paths from the source to the multicast members using a unicast QoS routing algorithm. Then, the obtained sub-graph can be reduced to optimize network utilization without violating the constraints. The aim of this reduction phase is to minimize the number of duplicated packets in links and to have a structure that approximates a tree structure as it is not necessarily the case. This strategy is adopted by the Multicast Adaptive Multiple Constraints Routing Algorithm (Mamcra) proposed in [11] as the multicast extension of Samcra, a unicast QoS routing algorithm [12] and the Taboo based QoS Multicast Routing algorithm (Taboo-QMR) [13].

III. THE MQMA ALGORITHM

In this section, we describe our proposed mQMA algorithm to achieve aggregated multi-constrained multicast routing.

A. Algorithm mechanisms

The mQMA algorithm achieves multicast tree aggregation in a centralized way by considering a centralized tree manager. The mQMA algorithm proceeds in three main steps as described below. When a new multicast group g arrives, the tree manager is contacted by the border router and after executing the first two steps, namely the computing path step and the tree decomposition, the tree manager tackles the aggregation step. In the following, we detail each of the three steps of the mQMA algorithm.

Step (1): Computing paths. The aim of this step is to compute multi-constrained paths from the source node to each destination node of the group. To perform this step, any unicast QoS routing algorithms can be used. In our simulation, we used the Samcra algorithm [12]. This first step is mandatory to determine one multi-constrained Feasible Paths Set (*FPS*) for a given group. The quality of the paths in *FPS* depends on the unicast routing algorithm used. If Samcra is used, feasible paths from source node to each destination node are found if they exist. The Samcra algorithm chooses the shortest path among all paths from the source to the destination according to the objective function used by Samcra (this function denoted by "Samcra length" is defined in section V). A destination (i.e.,

a group member) for which no path is found is then excluded, from the group as QoS requirements cannot be fulfilled. In this case, the *FPS* contains no path for this destination.

Step (2): Tree decomposition. The *FPS* obtained in the first step isn't necessarily a tree, it may contain cycles. The aim of this step is to decompose the *FPS* into several trees. The obtained trees will form the Feasible Trees Set (*FTS*). If the *FPS* contains no cycle, the *FTS* contains only one tree which corresponds to the graft of all the paths in the *FPS* and the algorithm goes directly to step 3. This tree decomposition works in a greedy manner by attempting to graft iteratively the paths for the members onto a tree of the *FTS*. If during an iteration, the graft of a path implies a cycle in all trees of the *FTS*, then this path is added to the *FTS* as a new tree. To proceed greedily, the *FPS* must be arranged according to a certain order criterion. In this paper and in the simulations, paths of the *FPS* were arranged according to the number of members. So, the path containing the most number of members is considered as the first tree in the *FTS* and then it is used to eventually graft the other paths of the *FPS*. The effect of adopting different order criteria can be explored. Another issue that can be investigated is the choice of the tree within the *FTS* to be used to graft paths when the path can be grafted on more than one tree. Actually, our algorithm selects the first tree in the *FTS* to which a new path can be aggregated. All these issues are beyond the scope of the present work.

Step (3): Aggregation.

After executing the first two steps, the tree manager attempts to achieve a global aggregation of the whole multicast group. It consists on finding from the *MTS* a tree MT_i that covers all the members covered by the *FPS* without violating the constraints and with a cost (in terms of the number of links) at most the cost of the *FPS*. If an MT_i exists, it is used by the multicast group G and all data for G are multiplexed onto that tree. If, however, several such MT_i can be used to aggregate G , the algorithm chooses the one of minimum cost. If such an MT_i doesn't exist and if the *FTS* contains only one tree, then the aggregation of G isn't possible and the tree in *FTS* must be added to the multicast tree set *MTS*. If the *FTS* contains more than one tree and if the global aggregation isn't possible, mQMA proceeds to a partial aggregation. Indeed, each tree t_i ($1 \leq i \leq |FTS|$) of the *FTS* covers a sub-group G_i of the initial multicast group G . The mQMA algorithm attempts to aggregate all the obtained sub-groups and if the aggregation is not possible, the corresponding tree is added to the multicast tree set (*MTS*) until all the members of G are satisfied. In this case, a multicast group is associated to several trees and the tree manager is in charge to add information that allows routers to multiplex data onto these trees.

B. mQMA on an example

Let us consider the topology presented in Figure 1 composed of 7 nodes and 7 links where s is the source node and nodes d_1 and d_2 are the members of the multicast group. We suppose that two additive metrics are considered, the values of which are portrayed aside to the different links. We consider

the constraint limits to be (6, 6). Execution of the first step of mQMA provides the set *FPS* of shortest paths to d_1 and d_2 . Here $FPS = \{FP_1 = (s-a-d_0-e-d_1), FP_2 = (s-b-d_0-e-d_2)\}$. During the second step of mQMA, the graph structure obtained as the union of paths in the *FPS* is decomposed into trees. The tree

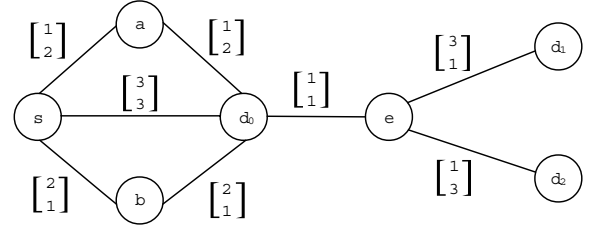


Fig. 1. mQMA : Network topology used to illustrate different mQMA mechanisms

decomposition step gives the set *FTS* composed of two trees, $FTS = \{FT_1 = \{(s-b-d_0-e-d_1)\}, FT_2 = \{(s-a-d_0-e-d_2)\}\}$.

The last step of mQMA, aims to find trees in the *MTS*, the multicast tree set, configured into the domain covering members and fulfilling the constraints. The global aggregation attempts to find one tree T to aggregate the entire group. If, it is not possible, mQMA attempts to aggregate the trees of the *FTS* tree per tree. Moreover, when paths are computed in the first step using a unicast QoS routing algorithm such as Samcra, only optimal paths are considered (optimal according to the multi-criteria length function defined by the algorithm). That's why during the aggregating step and according to the already accepted and configured groups, feasible but not optimal tree, covering the new coming group, may exist in the *MTS* and may be used to achieve global aggregation.

Let us consider another case where the constraint limits for a new request are fixed to (7, 7) and let us suppose that the *MTS* contains already the tree $MT_1 = \{(s-d_0-e-d_1-d_2)\}$. This tree is feasible for the new request so it can be used to aggregate the group (d_1, d_2) . If global aggregation is not possible, partial aggregation is executed in order to find in the *MTS*, trees on which FT_1 and/or FT_2 can be aggregated. If the *MTS* does not contain any tree that can be used to aggregate one of *FTS* trees, then the considered tree is added to the *MTS*. So, to summarize this example, four scenarios are possible: (1) The *MTS* contains one tree covering the member of the group and fulfilling the constraint limits; (2) if not, the *MTS* contains two trees covering, each one, one of the *FTS* trees (FT_1 and FT_2); (3) if not, the *MTS* contains one tree that covers one of the two *FTS* trees (FT_1 or FT_2), so the second uncovered tree of the *FTS* must be added to the *MTS* and must be configured into the domain; (4) if the two trees of the *FTS* can not be matched to any of the *MTS* trees, FT_1 and FT_2 are both added to the *MTS* and must be configured into the domain.

IV. MQMA: A PROPOSAL TO ENHANCE MULTI-CONSTRAINED ROUTING STRUCTURE

mQMA algorithm performs multicast aggregation taking into account multiple members requirements. In this section, we detail the cycles problem and how mQMA contributes in the reduction of forwarding structure.

A. Problem statement

Multicast routing aims to forward multicast data efficiently by sending single packets through the shared links and duplicating them if it is necessary through a tree structure. When multi-constrained multicast routing is considered, this philosophy may be no longer respected. Indeed, multi-constrained multicast routing computes a routing structure that satisfies multiple additive QoS metrics and this structure is not necessarily a tree. Not to affect the multicast philosophy, multi-constrained proposal takes into account the reduction of the computed routing structure in order to eliminate as many redundancies as possible without violating the members requirements.

Cycles problem on an example: Figure 1 illustrates the cycle problem. If the constraint limits are (8, 8), optimal paths (FP_1 and FP_2) for d_1 and d_2 form the cycle $(s-a-d_0-b-s)$. When rerouting data for d_2 through FP_1 or data for d_1 through FP_2 , the cycle $(s-a-d_0-b-s)$ is removed leading to a better use of network resources. When the constraint limits are fixed to (6,6). The path $(s-b-d_0-e-d_1)$ does not respect the constraints for d_1 and the path $(s-a-d_0-e-d_2)$ does not respect the constraints for d_2 . Consequently, the structure used for the group of source s and of members d_1 and d_2 contains the cycle $(s-a-d_0-b-s)$ which can not be removed. In this case, d_0 receives packets twice, once from a and once from b and the link (d_0-e) transfers duplicated packets. A specific routing has to be in place as traditional IP routing cannot support that case.

B. mQMA contribution to solve cycles problem

Besides the reduction of routing entries, the algorithm mQMA can be considered as an alternative to solve the cycles problem. In addition to the aggregation benefit, the routing structure used by mQMA to deliver multi-constrained multicast data has been implicitly reduced.

mQMA removes cycles during the two last steps. (1) During the tree decomposition, some cycles can be removed. Indeed, the FTS grows in a greedy manner with the add of paths. When a path is added, it may cover some members not already considered. In that case, and if the path is feasible for these members, they won't be considered at all. If the paths for these members were forming some cycles, then with tree decomposition, these cycles are removed. (2) During the phase of aggregation, one global tree that is feasible and that covers the members can exist even if the structure FTS contains more than one tree. In that case, the cycles are removed from the group quite easily as global aggregation is made.

To apprehend how mQMA eliminates cycles, let's consider Figure 1 where s is the source node and nodes d_0 , d_1 and d_2 the multicast members.

If the constraints are fixed to (7,7), and if Samcra is used as the unicast QoS routing, the first step of mQMA algorithm returns the set $FPS=\{FP_1=(s-a-d_0-e-d_1), FP_0=(s-d_0), FP_2=(s-b-d_0-e-d_2)\}$. The paths in the FPS do not form a tree, it contains 3 cycles which are (s,a,d_0,s) , (s,b,d_0,s) and (s,a,b,d_0,s) .

Cycles reduction during the tree decomposition step: The decomposition step consider as a first path, the path containing the most member nodes. So, the first computed tree is $FT_1=\{s-a-d_0-e-d_1\}$. The path $FP_2=(s-a-d_0-e-d_2)$ can not be grafted into the FT_1 due to the cycle (s,a,b,d_0) , so it must be added into the FTS as FT_2 and finally the path $FP_0=(s-d_0)$ used to forward data to d_0 is already in FT_1 , so this path will not be added to the FTS as a third tree. Thanks to this step, the cycles (s,a,d_0) , (s,b,d_0) are removed and then the FTS contains only 2 trees. The third step in mQMA attempts to aggregate these two trees to exiting trees of MTS .

Cycles reduction during the aggregation step : If the MTS , contains the tree $\{s-d_0-e-d_1-d_2\}$ and as the constraint limits are fixed to (7,7), this tree can be used to aggregate the multicast group. So, the routing structure used to forward data is now a tree and does no longer contain cycles. If the constraints are fixed to (6,6), global aggregation is not possible. In fact, the path $(s-d_0-e-d_1)$ and the path $(s-d_0-e-d_2)$ are not within constraints and so the tree $\{s-a-d_0-e-d_1-d_2\}$ can not be used to aggregate the whole group.

V. SIMULATIONS

This section presents the results of our conducted simulations to compare our proposed algorithm mQMA to the Mamcra algorithm. In the first step of mQMA, we have to use a unicast QoS routing algorithm. In our experimental study, we adopt Samcra as the unicast QoS routing algorithm to compute the set of feasible paths FPS . The optimality is obtained according to the Samcra length function.

A. Parameters of the simulations

The simulations were ran on the Abilene network [14] which contains 11 routers and 14 links. In this network, 1 000 groups were generated randomly with size between 2 and 11 members. The groups were source-oriented and the source for these groups was chosen among four routers which are known as specific delivering sources. The links l on the network were given values w_i^l (for the i -th metric) randomly from 1 which was 5 during the simulations. We considered 2 additive metrics and we generated the constrained limits L_i (for the i -th metric) randomly for all the groups between 5 and 500. For mQMA, the bandwidth threshold t_b varies from 0% for perfect aggregation to 40% for leaky aggregation. We repeat each scenario 1000 times. The simulation tool used to achieve this simulation is a graph simulator developed at IRISA and can be found at [15].

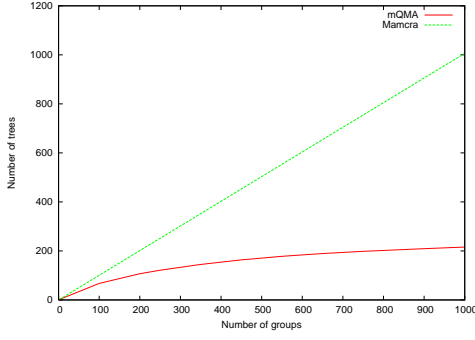


Fig. 2. Reduction of the total number of trees

B. Results of the simulations

This subsection presents the results of the simulations considering the number of trees, the network resource usage, the mean Samcra length per member and the number of cycles removed.

1) *Number of trees*: Figure 2 shows the reduction of the number of trees for mQMA. This performance is due to the multicast tree aggregation. We recall that Mamcra routing structure is not always a tree.

While Mamcra needs to maintain, as many structures as groups in the domain, (i.e. 1000), mQMA reduces the number of trees to only 200. As mQMA builds few trees, few forwarding states are maintained and the control messages to maintain the trees are reduced.

2) *Network resource usage*: Figure 3 shows the network resource usage in terms of relative utilization. This metric is denoted by

$$U_R = \frac{C(T_{mQMA}) - C(T_{Mamcra})}{C(T_{Mamcra})} = \frac{C(T_{mQMA})}{C(T_{Mamcra})} - 1$$

where $C(T_{algo})$ represents the cost of the routing structures of the algorithm *algo* used to forward data for all the generated groups. In the simulations, the cost is measured in terms of number of links used for each trees set. In other words, $C(T_{algo}) = \sum_{g \in \mathcal{G}} |t(g)|$, where \mathcal{G} represents the set of all the groups generated, $t(g)$ represents the structure used for g (which can be a set of trees), and $|t(g)|$ represents the number of links of $t(g)$. In mQMA, if the same tree is used to aggregate 2 groups, it is counted twice.

The relative utilization depends on the tolerated bandwidth threshold. When the bandwidth threshold exceeds 0%, mQMA can find an already existing tree that covers the members, that is feasible and that has a lower cost. This occurs when mQMA lists all the trees and chooses among the ones that are feasible, the tree that has the lowest number of links. The results show that even when leaky aggregation is adopted and for lower threshold ($\approx 17\%$), the aggregated trees used by mQMA to forward multicast traffic have lower cost than the Mamcra structure used when no aggregation is performed. So, even when leaky aggregation, mQMA may involve an effective gain in resource usage without violating the multi-constraint requirements of multicast groups. When the threshold increases, the network resource usage is not so high, the

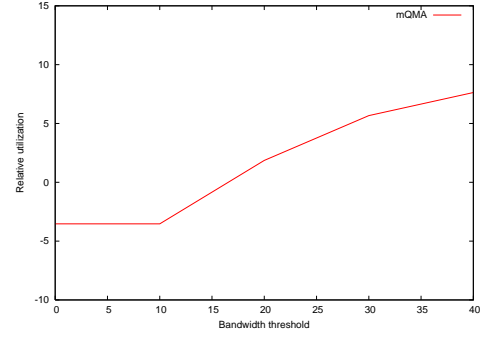


Fig. 3. Relative resource utilization : mQMA vs Mamcra

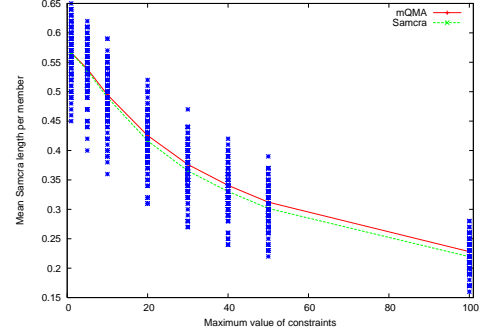


Fig. 4. Mean Samcra length per member

network resources waste for an infinite threshold does not exceed 35%.

3) *Mean Samcra length per member*: Figure 4 shows the mean Samcra length for the members, which is the main metric used for multi-constrained routing algorithm. This metric reflects the quality of the communications for the members of the group, expressed in function of the different QoS metrics. The Samcra length from s to d is denoted by :

$$Samcra\ length(s,d) = \max_{i < \# \text{ metrics}} \sum_{e_j \in \text{path}(s,d)} \frac{w_i^{e_j}}{L_i},$$

where $w_i^{e_j}$ is the value of the metric number i for the edge e_j ($e_j \in P$), and L_i is the requirement of the group for the metric i . The algorithm Samcra computes the optimal structure in minimizing the value of this metric. The results show that mQMA behaves in the same way as Samcra with slightly higher results for this metric. Then, we can say that the two algorithms are comparable and that mQMA gives good performance in terms of Samcra length as its results are close to the optimal.

4) *Number of cycles removed*: To achieve multi-constrained multicast routing, the routing structure is not always tree as it is detailed in section IV. The graph structure computed by the unicast QoS routing algorithm may contain cycles. Figure 5 illustrates the number of cases with cycles contained on the structure computed by the unicast algorithm to reach all the multicast members (namely Samcra). It represents the number of cases in which the set FPS of feasible paths contains cycles. Figure 5 compares the number of cases in which

	Structures with cycles	Without Mamcra	Without cycles -	Without mQMA	Without cycles -	Without cycles - Tree Decomposition
Maximum Value metric = 5	22402	21559		21754		15757
Total number of cycles removed (in %)		95.59% of cycles removed		96.84% of cycles removed		68.72% of cycles removed

TABLE I
MQMA REMOVES MORE CYCLES FROM THE STRUCTURES THAN MAMCRA DOES

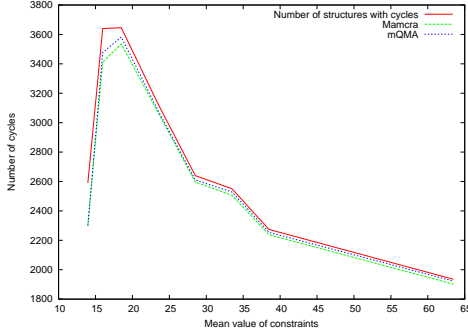


Fig. 5. Number of cycles removed

mQMA eliminates cycles to the number of cases in which Mamcra succeed to do it. We recall, that cycles reduction obtained by mQMA include the one achieved by the tree decomposition and the one achieved by the aggregation. In some cases, neither Mamcra nor mQMA removes the cycles found in *FPS*. In most of these cases, the cycles can not be removed and the structure *FPS* is the only one possible that respects the constraints fixed for the group. The results show that the two algorithms are comparable in terms of cycle reduction as they behave in the same way. The table I shows the number of cycles removed by Mamcra, mQMA and tree decomposition when generating 30000 test cases. In 74% of the generated cases, the structure *FPS* found by Samcra contains at least one cycle. This confirms the necessity of finding an algorithm that removes the cycles. At the end of the simulations, mQMA has removed slightly more trees than Mamcra, that means that in 95.59% of cases, Mamcra has removed all the cycles from the *FPS* while mQMA has removed all the cycles in 96.84% of cases. From these 96.84% cases obtained by mQMA, 68.72% are implied by the tree decomposition step. The other remaining cases are implied by global aggregation.

VI. CONCLUSION

We presented in this paper a new algorithm mQMA that deals with the two main problems of multicast deployment: multicast forwarding state scalability and multi-constrained QoS routing. Our protocol is based on the techniques of both multicast aggregation and multi-constrained routing. To the best of our knowledge, mQMA is the first protocol that deals with these two main problems. With mQMA, in the worst

case, a group can be associated to several trees and the data for the group are multiplexed onto these trees.

Conducted simulations showed that our algorithm achieves indeed comparable performances in terms of Samcra length as Samcra itself which provides the optimal routes for this metric. Moreover, mQMA spares network resources by using much less than Mamcra. Finally, mQMA gains all the benefits of the multicast tree aggregation by building very few trees (at maximum 200 trees for 1 000 groups for Abilene network) and maintaining consequently few forwarding states for these groups.

REFERENCES

- [1] J.-H. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla, "Aggregated Multicast — A Comparative Study," *Special issue of Cluster Computing: The Journal of Networks, Software and Applications*, 2003.
- [2] A. Guitton and J. Moulrierac, "Scalable Tree Aggregation for Multicast," in *8th International Conference on Telecommunications (ConTEL)*, June 2005, best student paper award.
- [3] J.-H. Cui, L. Lao, D. Maggiorini, and M. Gerla, "BEAM: A Distributed Aggregated Multicast Protocol Using Bi-directional Trees," in *IEEE International Conference on Communications (ICC)*, May 2003.
- [4] Z.-F. Liu, W.-H. Dou, and Y.-J. Liu, "AMBTs: A Scheme of Aggregated Multicast Based on Tree Splitting," in *IFIP Networking*, ser. LNCS, no. 3042, May 2004, pp. 829–840.
- [5] J. Moulrierac, A. Guitton, and M. Molnár, "Multicast Tree Aggregation in Large Domains," in *IFIP Networking*, 2006.
- [6] J.-H. Cui, L. Lao, M. Faloutsos, and M. Gerla, "AQoSM: Scalable QoS multicast provisioning in Diff-Serv networks," *Computer Networks*, 2005.
- [7] J. Moulrierac and A. Guitton, "QoS Scalable Tree Aggregation," in *IFIP Networking*, ser. LNCS, no. 3462, May 2005, pp. 1405–1408.
- [8] S. Chen and K. Nahrstedt, "An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions," *IEEE Network, Special Issue on Transmission and Distribution of digital Video*, pp. 64–79, November/December 1998.
- [9] J. Moulrierac, A. Guitton, and M. Molnár, "On the number of MPLS LSP using Multicast Tree Aggregation," in *IEEE Globecom*, November 2006.
- [10] P. Van Mieghem, F. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-Bruin, J. Solé-Pareta, and S. Sánchez-López, *Quality of Future Internet Services*, smirnov and al. ed., ser. EU-COST 263 Final Report. Springer LNCS 2856, 2003, ch. 3, pp. 80–117.
- [11] F. Kuipers and V. M. P., "Constrained-based multicast routing algorithm," *Computer Communications*, vol. 25, no. 8, pp. 801–810, 2002.
- [12] P. Van Mieghem, H. De Neve, and F. Kuipers, "Hop-by-hop quality of service routing," *Computer Networks*, vol. 37, no. 3–4, pp. 407–423, 2001.
- [13] N. Ben Ali, M. Molnar, and A. Belghith, "Résolution du probleme de routage multicast multicriterie," in *Metaheuristics Workshop*, 2006.
- [14] Abilene network, "http://abilene.internet2.edu."
- [15] Graph Simulator, "http://www.irisa.fr/armor/lesmembres/Moulrierac/outil.html."